

Informe de Laboratorio 04

Modelo de Datos de la Aplicación EMPLEOYA (Django)

Nota

Estudiantes	Escuela	Asignatura
Piero De La Cruz Mancilla , Jerson Ernesto Chura Pacci pdelacruz@ulasalle.edu.pe, jchurap@ulasalle.edu.pe	Carrera Profesional de Ingeniería de Software	Ingeniería Web Semestre: VII VII

Laboratorio	Tema	Duración
04	Modelo de Datos de la Aplicación EMPLEOYA (Django)	1 semana

1. Introducción

El presente informe documenta el **modelo de datos** de la aplicación **EMPLEOYA**. El modelo fue implementado con el framework **Django** y está orientado a gestionar usuarios, empresas, ofertas laborales, postulaciones, favoritos y notificaciones.

Este documento incluye:

- Descripción del dominio y objetivos del modelo.
- Código fuente relevante (archivo `models.py`) con comentarios y explicación por entidad.
- Migración inicial (`0001_initial.py`) y explicación de la traducción a tablas físicas.
- Consideraciones de diseño, índices y restricciones, ejemplos de consultas ORM y comandos para pruebas.

2. Objetivos

- Definir el modelo de datos necesario para la aplicación EMPLEOYA.
- Implementar el modelo en Django siguiendo buenas prácticas.
- Documentar las decisiones de diseño, índices y relaciones entre entidades.
- Proveer instrucciones mínimas para ejecutar y probar el esquema en un entorno local.

3. Alcance

Este informe cubre el diseño y la implementación del modelo de datos. No incluye la descripción completa de la UI, pruebas unitarias completas, ni el código de todas las vistas. Se documentan los componentes del dominio y cómo se materializan en la base de datos.

4. Modelo Conceptual y Entidades

Las entidades cubiertas por el modelo son:

- **Usuario** (roles: postulante, empleador, admin).
- **Categoria** (clasificación de ofertas).
- **Empresa** (perfil del empleador).
- **PerfilPostulante** (datos curriculares del postulante).
- **OfertaTrabajo** (vacantes).
- **Postulacion** (aplicación de postulante a oferta).
- **Favorito** (marcador de oferta por usuario).
- **Notificacion** (mensajes del sistema a usuarios).

5. Implementación: Código fuente (lo más importante)

Abajo se inserta el contenido esencial del archivo `models.py` que entregaste. Cada bloque incluye explicación corta sobre su responsabilidad y decisiones importantes. Mantengo exactamente la semántica y campos que entregaste.

5.1. Archivo: `models.py`

Descripción: define los modelos que representan las tablas principales del dominio. Incluye un manager personalizado para la entidad **Usuario** y validaciones básicas.

Listing 1: `models.py` - Definición completa (proporcionada por el equipo)

```
1 from django.db import models
2 from django.contrib.auth.models import AbstractBaseUser, BaseUserManager, PermissionsMixin
3 from django.core.validators import MinValueValidator, MaxValueValidator
4 from django.utils import timezone
5
6 # Modelos para el sistema EMPLEOYA
7 # Aqu definimos las tablas de la base de datos
8
9 class UsuarioManager(BaseUserManager):
10     """Manager para crear usuarios - necesario para usar email en vez de username"""
11
12     def create_user(self, email, password=None, **extra_fields):
13         """Crear y guardar un usuario regular"""
14         if not email:
15             raise ValueError('El email es obligatorio')
16
17         email = self.normalize_email(email)
18         user = self.model(email=email, **extra_fields)
19         user.set_password(password)
20         user.save(using=self._db)
21         return user
22
23     def create_superuser(self, email, password=None, **extra_fields):
24         """Crear y guardar un superusuario"""
25         extra_fields.setdefault('is_staff', True)
26         extra_fields.setdefault('is_superuser', True)
27         extra_fields.setdefault('tipo_usuario', 'admin')
28
29         return self.create_user(email, password, **extra_fields)
30
31
```

```
32 class Usuario(AbstractBaseUser, PermissionsMixin):
33     """Modelo de Usuario para EMPLEOYA"""
34
35     TIPO_USUARIO_CHOICES = [
36         ('postulante', 'Postulante'),
37         ('empleador', 'Empleador'),
38         ('admin', 'Administrador'),
39     ]
40
41     ESTADO_CHOICES = [
42         ('activo', 'Activo'),
43         ('inactivo', 'Inactivo'),
44         ('suspendido', 'Suspendido'),
45     ]
46
47     email = models.EmailField(unique=True, verbose_name='Correo Electrónico')
48     nombre = models.CharField(max_length=100, verbose_name='Nombre')
49     apellido = models.CharField(max_length=100, blank=True, verbose_name='Apellido')
50     telefono = models.CharField(max_length=20, blank=True, null=True, verbose_name='Teléfono')
51     tipo_usuario = models.CharField(
52         max_length=20,
53         choices=TIPO_USUARIO_CHOICES,
54         default='postulante',
55         verbose_name='Tipo de Usuario'
56     )
57     estado = models.CharField(
58         max_length=20,
59         choices=ESTADO_CHOICES,
60         default='activo',
61         verbose_name='Estado'
62     )
63     email_verificado = models.BooleanField(default=False, verbose_name='Email Verificado')
64     fecha_creacion = models.DateTimeField(auto_now_add=True, verbose_name='Fecha de Creación')
65     fecha_actualizacion = models.DateTimeField(auto_now=True, verbose_name='Última Actualización')
66
67     # Campos requeridos por Django
68     is_active = models.BooleanField(default=True)
69     is_staff = models.BooleanField(default=False)
70
71     objects = UsuarioManager()
72
73     USERNAME_FIELD = 'email'
74     REQUIRED_FIELDS = ['nombre']
75
76     class Meta:
77         db_table = 'usuario'
78         verbose_name = 'Usuario'
79         verbose_name_plural = 'Usuarios'
80         ordering = ['-fecha_creacion']
81
82     def __str__(self):
83         return f"{self.nombre} {self.apellido} ({self.email})"
84
85     @property
86     def nombre_completo(self):
87         return f"{self.nombre} {self.apellido}".strip()
88
89
90 class Categoria(models.Model):
91     """Categorías de ofertas de trabajo"""
92
93     nombre = models.CharField(max_length=100, unique=True, verbose_name='Nombre')
94     descripcion = models.TextField(blank=True, null=True, verbose_name='Descripción')
95     icono = models.CharField(max_length=50, blank=True, null=True, verbose_name='Icono')
96     activa = models.BooleanField(default=True, verbose_name='Activa')
97     fecha_creacion = models.DateTimeField(auto_now_add=True, verbose_name='Fecha de Creación')
98
99     class Meta:
100         db_table = 'categoria'
101         verbose_name = 'Categoría'
102         verbose_name_plural = 'Categorías'
103         ordering = ['nombre']
104
105     def __str__(self):
106         return self.nombre
107
108
109 class Empresa(models.Model):
110     """Perfil de empresa/empleador"""
111
112     TAMANO_EMPRESA_CHOICES = [
113         ('startup', 'Startup (1-10)'),
114         ('pyme', 'PYME (11-50)'),
115         ('mediana', 'Mediana (51-200)'),
116         ('grande', 'Grande (201-1000)'),
117         ('corporacion', 'Corporación (1000+)'),
118     ]
119
120     usuario = models.OneToOneField(
121         Usuario,
122         on_delete=models.CASCADE,
123         related_name='empresa',
124         verbose_name='Usuario'
125     )
126     nombre_empresa = models.CharField(max_length=200, verbose_name='Nombre de la Empresa')
127     ruc = models.CharField(max_length=20, unique=True, null=True, blank=True, verbose_name='RUC')
128     descripcion = models.TextField(blank=True, null=True, verbose_name='Descripción')
129     sector = models.CharField(max_length=100, blank=True, null=True, verbose_name='Sector')
130     ubicacion = models.CharField(max_length=200, blank=True, null=True, verbose_name='Ubicación')
131     sitio_web = models.URLField(blank=True, null=True, verbose_name='Sitio Web')
132     logo_url = models.CharField(max_length=500, blank=True, null=True, verbose_name='URL del Logo')
133     tamano_empresa = models.CharField(
134         max_length=20,
```

```
135     choices=TAMANO_EMPRESA_CHOICES,
136     default='pyme',
137     verbose_name='Tamaño de Empresa'
138 )
139 telefono_empresa = models.CharField(max_length=20, blank=True, null=True, verbose_name='Telefono')
140 verificada = models.BooleanField(default=False, verbose_name='Verificada')
141 fecha_creacion = models.DateTimeField(auto_now_add=True, verbose_name='Fecha de Creación')
142 fecha_actualizacion = models.DateTimeField(auto_now=True, verbose_name='Última Actualización')
143
144 class Meta:
145     db_table = 'empresa'
146     verbose_name = 'Empresa'
147     verbose_name_plural = 'Empresas'
148     ordering = ['-fecha_creacion']
149
150 def __str__(self):
151     return self.nombre_empresa
152
153
154 class PerfilPostulante(models.Model):
155     """Perfil del postulante"""
156
157     NIVEL_EXPERIENCIA_CHOICES = [
158         ('sin_experiencia', 'Sin Experiencia'),
159         ('junior', 'Junior (1-2 años)'),
160         ('semi_senior', 'Semi Senior (3-5 años)'),
161         ('senior', 'Senior (6-10 años)'),
162         ('lead', 'Lead/Experto (10+ años)'),
163     ]
164
165     DISPONIBILIDAD_CHOICES = [
166         ('inmediata', 'Inmediata'),
167         ('2_semanas', '2 Semanas'),
168         ('1_mes', '1 Mes'),
169         ('negociable', 'Negociable'),
170     ]
171
172     usuario = models.OneToOneField(
173         Usuario,
174         on_delete=models.CASCADE,
175         related_name='perfil_postulante',
176         verbose_name='Usuario'
177     )
178     titulo_profesional = models.CharField(max_length=200, blank=True, null=True, verbose_name='Título Profesional')
179     resumen_profesional = models.TextField(blank=True, null=True, verbose_name='Resumen Profesional')
180     nivel_experiencia = models.CharField(
181         max_length=20,
182         choices=NIVEL_EXPERIENCIA_CHOICES,
183         default='sin_experiencia',
184         verbose_name='Nivel de Experiencia'
185     )
186     aos_experiencia = models.IntegerField(
187         default=0,
188         validators=[MinValueValidator(0), MaxValueValidator(50)],
189         verbose_name='Años de Experiencia'
190     )
191     habilidades = models.TextField(blank=True, null=True, verbose_name='Habilidades')
192     educacion = models.TextField(blank=True, null=True, verbose_name='Educación')
193     experiencia_laboral = models.TextField(blank=True, null=True, verbose_name='Experiencia Laboral')
194     certificaciones = models.TextField(blank=True, null=True, verbose_name='Certificaciones')
195     idiomas = models.TextField(blank=True, null=True, verbose_name='Idiomas')
196     cv_url = models.CharField(max_length=500, blank=True, null=True, verbose_name='URL del CV')
197     foto_perfil_url = models.CharField(max_length=500, blank=True, null=True, verbose_name='URL Foto de Perfil')
198     ubicacion = models.CharField(max_length=200, blank=True, null=True, verbose_name='Ubicación')
199     salario_esperado = models.DecimalField(
200         max_digits=10,
201         decimal_places=2,
202         blank=True,
203         null=True,
204         verbose_name='Salario Esperado'
205     )
206     moneda_salario = models.CharField(max_length=3, default='PEN', verbose_name='Moneda')
207     disponibilidad = models.CharField(
208         max_length=20,
209         choices=DISPONIBILIDAD_CHOICES,
210         default='negociable',
211         verbose_name='Disponibilidad'
212     )
213     portafolio_url = models.URLField(blank=True, null=True, verbose_name='URL del Portafolio')
214     linkedin_url = models.URLField(blank=True, null=True, verbose_name='URL de LinkedIn')
215     github_url = models.URLField(blank=True, null=True, verbose_name='URL de GitHub')
216     completado = models.BooleanField(default=False, verbose_name='Perfil Completado')
217     fecha_creacion = models.DateTimeField(auto_now_add=True, verbose_name='Fecha de Creación')
218     fecha_actualizacion = models.DateTimeField(auto_now=True, verbose_name='Última Actualización')
219
220 class Meta:
221     db_table = 'perfil_postulante'
222     verbose_name = 'Perfil de Postulante'
223     verbose_name_plural = 'Perfiles de Postulantes'
224     ordering = ['-fecha_actualizacion']
225
226 def __str__(self):
227     return f'{self.usuario.nombre_completo} - {self.titulo_profesional or "Sin título"}'
228
229
230 class OfertaTrabajo(models.Model):
231     """Ofertas de trabajo publicadas por empresas"""
232
233     MODALIDAD_CHOICES = [
234         ('presencial', 'Presencial'),
235         ('remoto', 'Remoto'),
236         ('hibrido', 'Híbrido'),
237     ]
```

```
238
239
240 TIPO_CONTRATO_CHOICES = [
241     ('tiempo_completo', 'Tiempo Completo'),
242     ('medio_tiempo', 'Medio Tiempo'),
243     ('por_proyecto', 'Por Proyecto'),
244     ('freelance', 'Freelance'),
245     ('practicas', 'Prcticas'),
246     ('temporal', 'Temporal'),
247 ]
248
249 NIVEL_EXPERIENCIA_CHOICES = [
250     ('sin_experiencia', 'Sin Experiencia'),
251     ('junior', 'Junior'),
252     ('semi_senior', 'Semi Senior'),
253     ('senior', 'Senior'),
254     ('lead', 'Lead/Experto'),
255 ]
256
257 ESTADO_CHOICES = [
258     ('borrador', 'Borrador'),
259     ('pendiente_aprobacion', 'Pendiente de Aprobacin'),
260     ('activa', 'Activa'),
261     ('pausada', 'Pausada'),
262     ('expirada', 'Expirada'),
263     ('cerrada', 'Cerrada'),
264 ]
265
266 empresa = models.ForeignKey(
267     Empresa,
268     on_delete=models.CASCADE,
269     related_name='ofertas',
270     verbose_name='Empresa'
271 )
272
273 categoria = models.ForeignKey(
274     Categoria,
275     on_delete=models.SET_NULL,
276     null=True,
277     related_name='ofertas',
278     verbose_name='Categoría'
279 )
280
281 titulo = models.CharField(max_length=200, verbose_name='Título')
282 descripcion = models.TextField(verbose_name='Descripción')
283 requisitos = models.TextField(blank=True, null=True, verbose_name='Requisitos')
284 responsabilidades = models.TextField(blank=True, null=True, verbose_name='Responsabilidades')
285 beneficios = models.TextField(blank=True, null=True, verbose_name='Beneficios')
286 salario_min = models.DecimalField(
287     max_digits=10,
288     decimal_places=2,
289     blank=True,
290     null=True,
291     verbose_name='Salario Mímo'
292 )
293
294 salario_max = models.DecimalField(
295     max_digits=10,
296     decimal_places=2,
297     blank=True,
298     null=True,
299     verbose_name='Salario Máximo'
300 )
301
302 moneda = models.CharField(max_length=3, default='PEN', verbose_name='Moneda')
303 ubicacion = models.CharField(max_length=200, blank=True, null=True, verbose_name='Ubicación')
304 modalidad = models.CharField(
305     max_length=20,
306     choices=MODALIDAD_CHOICES,
307     verbose_name='Modalidad'
308 )
309
310 tipo_contrato = models.CharField(
311     max_length=20,
312     choices=TIPO_CONTRATO_CHOICES,
313     verbose_name='Tipo de Contrato'
314 )
315
316 nivel_experiencia = models.CharField(
317     max_length=20,
318     choices=NIVEL_EXPERIENCIA_CHOICES,
319     verbose_name='Nivel de Experiencia Requerido'
320 )
321
322 vacantes_disponibles = models.IntegerField(
323     default=1,
324     validators=[MinValueValidator(1)],
325     verbose_name='Vacantes Disponibles'
326 )
327
328 fecha_publicacion = models.DateTimeField(blank=True, null=True, verbose_name='Fecha de Publicación')
329 fecha_expiracion = models.DateTimeField(blank=True, null=True, verbose_name='Fecha de Expiración')
330 fecha_inicio_deseada = models.DateField(blank=True, null=True, verbose_name='Fecha de Inicio Deseada')
331 estado = models.CharField(
332     max_length=30,
333     choices=ESTADO_CHOICES,
334     default='borrador',
335     verbose_name='Estado'
336 )
337
338 aprobada_admin = models.BooleanField(default=False, verbose_name='Aprobada por Admin')
339 fecha_aprobacion = models.DateTimeField(blank=True, null=True, verbose_name='Fecha de Aprobación')
340 vistas = models.IntegerField(default=0, verbose_name='Número de Vistas')
341 fecha_creacion = models.DateTimeField(auto_now_add=True, verbose_name='Fecha de Creación')
342 fecha_actualizacion = models.DateTimeField(auto_now=True, verbose_name='Última Actualización')
343
344 class Meta:
345     db_table = 'oferta_trabajo'
346     verbose_name = 'Oferta de Trabajo'
347     verbose_name_plural = 'Ofertas de Trabajo'
348     ordering = ['-fecha_publicacion', '-fecha_creacion']
349     indexes = [
350         models.Index(fields=['estado', 'fecha_publicacion']),
351     ]
```

```
341         models.Index(fields=['categoria', 'estado']),
342         models.Index(fields=['modalidad', 'estado']),
343     ]
344
345     def __str__(self):
346         return f"{self.titulo} - {self.empresa.nombre_empresa}"
347
348     def save(self, *args, **kwargs):
349         # Si la oferta se activa y no tiene fecha de publicacin, asignarla
350         if self.estado == 'activa' and not self.fecha_publicacion:
351             self.fecha_publicacion = timezone.now()
352         super().save(*args, **kwargs)
353
354
355 class Postulacion(models.Model):
356     """Postulaciones de candidatos a ofertas de trabajo"""
357
358     ESTADO_CHOICES = [
359         ('pendiente', 'Pendiente'),
360         ('en_revision', 'En Revisin'),
361         ('preseleccionado', 'Preseleccionado'),
362         ('entrevista', 'En Entrevista'),
363         ('rechazado', 'Rechazado'),
364         ('aceptado', 'Aceptado'),
365     ]
366
367     oferta = models.ForeignKey(
368         OfertaTrabajo,
369         on_delete=models.CASCADE,
370         related_name='postulaciones',
371         verbose_name='Oferta de Trabajo'
372     )
373     postulante = models.ForeignKey(
374         PerfilPostulante,
375         on_delete=models.CASCADE,
376         related_name='postulaciones',
377         verbose_name='Postulante'
378     )
379     fecha_postulacion = models.DateTimeField(auto_now_add=True, verbose_name='Fecha de Postulacin')
380     estado = models.CharField(
381         max_length=20,
382         choices=ESTADO_CHOICES,
383         default='pendiente',
384         verbose_name='Estado'
385     )
386     carta_presentacion = models.TextField(blank=True, null=True, verbose_name='Carta de Presentacin')
387     cv_url_postulacion = models.CharField(max_length=500, blank=True, null=True, verbose_name='URL del CV')
388     fecha_cambio_estado = models.DateTimeField(blank=True, null=True, verbose_name='Fecha de Cambio de Estado')
389     notas_empleador = models.TextField(blank=True, null=True, verbose_name='Notas del Empleador')
390     puntuacion_match = models.IntegerField(
391         blank=True,
392         null=True,
393         validators=[MinValueValidator(0), MaxValueValidator(100)],
394         verbose_name='Puntuacin de Match (%)'
395     )
396
397     class Meta:
398         db_table = 'postulacion'
399         verbose_name = 'Postulacin'
400         verbose_name_plural = 'Postulaciones'
401         ordering = ['-fecha_postulacion']
402         unique_together = ['oferta', 'postulante']
403         indexes = [
404             models.Index(fields=['estado', 'fecha_postulacion']),
405             models.Index(fields=['oferta', 'estado']),
406             models.Index(fields=['postulante', 'estado']),
407         ]
408
409     def __str__(self):
410         return f"{self.postulante.usuario.nombre_completo} -> {self.oferta.titulo}"
411
412     def save(self, *args, **kwargs):
413         # Actualizar fecha de cambio de estado si cambi el estado
414         if self.pk:
415             old_instance = Postulacion.objects.get(pk=self.pk)
416             if old_instance.estado != self.estado:
417                 self.fecha_cambio_estado = timezone.now()
418         super().save(*args, **kwargs)
419
420
421 class Favorito(models.Model):
422     """Ofertas marcadas como favoritas por postulantes"""
423
424     usuario = models.ForeignKey(
425         Usuario,
426         on_delete=models.CASCADE,
427         related_name='favoritos',
428         verbose_name='Usuario'
429     )
430     oferta = models.ForeignKey(
431         OfertaTrabajo,
432         on_delete=models.CASCADE,
433         related_name='favoritos',
434         verbose_name='Oferta'
435     )
436     fecha_agregado = models.DateTimeField(auto_now_add=True, verbose_name='Fecha Agregado')
437
438     class Meta:
439         db_table = 'favorito'
440         verbose_name = 'Favorito'
441         verbose_name_plural = 'Favoritos'
442         unique_together = ['usuario', 'oferta']
443         ordering = ['-fecha_agregado']
```

```

444
445     def __str__(self):
446         return f"{self.usuario.nombre_completo} - {self.oferta.titulo}"
447
448
449 class Notificacion(models.Model):
450     """Notificaciones para usuarios"""
451
452     TIPO_CHOICES = [
453         ('postulacion', 'Nueva Postulacin'),
454         ('estado_postulacion', 'Cambio Estado Postulacin'),
455         ('nueva_oferta', 'Nueva Oferta'),
456         ('mensaje', 'Mensaje'),
457         ('alerta', 'Alerta'),
458         ('sistema', 'Sistema'),
459     ]
460
461     usuario = models.ForeignKey(
462         Usuario,
463         on_delete=models.CASCADE,
464         related_name='notificaciones',
465         verbose_name='Usuario'
466     )
467     tipo = models.CharField(max_length=30, choices=TIPO_CHOICES, verbose_name='Tipo')
468     titulo = models.CharField(max_length=200, verbose_name='Titulo')
469     mensaje = models.TextField(verbose_name='Mensaje')
470     enlace = models.CharField(max_length=500, blank=True, null=True, verbose_name='Enlace')
471     leida = models.BooleanField(default=False, verbose_name='Leda')
472     fecha_creacion = models.DateTimeField(auto_now_add=True, verbose_name='Fecha de Creacin')
473     fecha_leida = models.DateTimeField(blank=True, null=True, verbose_name='Fecha de Lectura')
474
475     class Meta:
476         db_table = 'notificacion'
477         verbose_name = 'Notificacion'
478         verbose_name_plural = 'Notificaciones'
479         ordering = ['-fecha_creacion']
480         indexes = [
481             models.Index(fields=['usuario', 'leida', '-fecha_creacion']),
482         ]
483
484     def __str__(self):
485         return f"{self.tipo} - {self.usuario.email} - {self.titulo}"

```

Notas sobre models.py

- **UsuarioManager**: permite crear usuarios usando email como identificador (`USERNAME_FIELD = 'email'`). `create_superuser('marcais_taff/isuperuser')`.
- **OneToOneField** en Empresa y PerfilPostulante garantiza que un usuario tenga como máximo un perfil de cada tipo.
- **ForeignKey(on_delete=CASCADE)** en ofertas y postulaciones modela eliminación en cascada cuando la entidad padre se borra.
- Índices declarados en **OfertaTrabajo** y **Postulacion** aceleran consultas por estado y fechas, que son las más frecuentes en el sistema.
- **unique_together** en **Postulacion** y **Favorito** evita duplicados lógicos.
- Campos tipo URL y campos de texto largo permiten almacenar enlaces a CV/portafolio sin incorporar archivos binarios en la base.

6. Migración inicial (archivo proporcionado: 0001_initial.py)

El siguiente bloque es la migración inicial auto-generada por Django. Contiene la creación de tablas, índices y restricciones tal como aparecen en la base de datos.

Listing 2: 0001_initial.py — Migración inicial (proporcionada)

```

1 # Generated by Django 5.2.7 on 2025-10-06 02:53
2
3 import django.core.validators
4 import django.db.models.deletion
5 from django.conf import settings
6 from django.db import migrations, models
7
8
9 class Migration(migrations.Migration):

```

```
10 initial = True
11
12
13 dependencies = [
14     ('auth', '0012_alter_user_first_name_max_length'),
15 ]
16
17 operations = [
18     migrations.CreateModel(
19         name='Categoria',
20         fields=[
21             ('id', models.BigAutoField(auto_created=True, primary_key=True, serialize=False, verbose_name='ID')),
22             ('nombre', models.CharField(max_length=100, unique=True, verbose_name='Nombre')),
23             ('descripcion', models.TextField(blank=True, null=True, verbose_name='Descripción')),
24             ('icono', models.CharField(blank=True, max_length=50, null=True, verbose_name='Icono')),
25             ('activa', models.BooleanField(default=True, verbose_name='Activa')),
26             ('fecha_creacion', models.DateTimeField(auto_now_add=True, verbose_name='Fecha de Creación')),
27         ],
28         options={
29             'verbose_name': 'Categoría',
30             'verbose_name_plural': 'Categorías',
31             'db_table': 'categoria',
32             'ordering': ['nombre'],
33         },
34     ),
35     migrations.CreateModel(
36         name='Empresa',
37         fields=[
38             ('id', models.BigAutoField(auto_created=True, primary_key=True, serialize=False, verbose_name='ID')),
39             ('nombre_empresa', models.CharField(max_length=200, verbose_name='Nombre de la Empresa')),
40             ('ruc', models.CharField(blank=True, max_length=20, null=True, unique=True, verbose_name='RUC')),
41             ('descripcion', models.TextField(blank=True, null=True, verbose_name='Descripción')),
42             ('sector', models.CharField(blank=True, max_length=100, null=True, verbose_name='Sector')),
43             ('ubicacion', models.CharField(blank=True, max_length=200, null=True, verbose_name='Ubicación')),
44             ('sitio_web', models.URLField(blank=True, null=True, verbose_name='Sitio Web')),
45             ('logo_url', models.CharField(blank=True, max_length=500, null=True, verbose_name='URL del Logo')),
46             ('tamao_empresa', models.CharField(choices=[('startup', 'Startup (1-10)'), ('pyme', 'PYME (11-50)'), ('mediana', 'Mediana (51-200)'), ('grande', 'Grande (201-1000)'), ('corporacion', 'Corporación (1000+)')], default='pyme', max_length=20, verbose_name='Tamaño de Empresa')),
47             ('telefono_empresa', models.CharField(blank=True, max_length=20, null=True, verbose_name='Teléfono')),
48             ('verificada', models.BooleanField(default=False, verbose_name='Verificada')),
49             ('fecha_creacion', models.DateTimeField(auto_now_add=True, verbose_name='Fecha de Creación')),
50             ('fecha_actualizacion', models.DateTimeField(auto_now=True, verbose_name='Última Actualización')),
51         ],
52         options={
53             'verbose_name': 'Empresa',
54             'verbose_name_plural': 'Empresas',
55             'db_table': 'empresa',
56             'ordering': ['-fecha_creacion'],
57         },
58     ),
59     migrations.CreateModel(
60         name='PerfilPostulante',
61         fields=[
62             ('id', models.BigAutoField(auto_created=True, primary_key=True, serialize=False, verbose_name='ID')),
63             ('titulo_profesional', models.CharField(blank=True, max_length=200, null=True, verbose_name='Título Profesional')),
64             ('resumen_profesional', models.TextField(blank=True, null=True, verbose_name='Resumen Profesional')),
65             ('nivel_experiencia', models.CharField(choices=[('sin_experiencia', 'Sin Experiencia'), ('junior', 'Junior (1-2 años)'), ('semi_senior', 'Semi Senior (3-5 años)'), ('senior', 'Senior (6-10 años)'), ('lead', 'Lead/Experto (10+ años)')], default='sin_experiencia', max_length=20, verbose_name='Nivel de Experiencia')),
66             ('años_experiencia', models.IntegerField(default=0, validators=[django.core.validators.MinValueValidator(0), django.core.validators.MaxValueValidator(50)], verbose_name='Años de Experiencia')),
67             ('habilidades', models.TextField(blank=True, null=True, verbose_name='Habilidades')),
68             ('educacion', models.TextField(blank=True, null=True, verbose_name='Educación')),
69             ('experiencia_laboral', models.TextField(blank=True, null=True, verbose_name='Experiencia Laboral')),
70             ('certificaciones', models.TextField(blank=True, null=True, verbose_name='Certificaciones')),
71             ('idiomas', models.TextField(blank=True, null=True, verbose_name='Idiomas')),
72             ('cv_url', models.CharField(blank=True, max_length=500, null=True, verbose_name='URL del CV')),
73             ('foto_perfil_url', models.CharField(blank=True, max_length=500, null=True, verbose_name='URL Foto de Perfil')),
74             ('ubicacion', models.CharField(blank=True, max_length=200, null=True, verbose_name='Ubicación')),
75             ('salario_esperado', models.DecimalField(blank=True, decimal_places=2, max_digits=10, null=True, verbose_name='Salario Esperado')),
76             ('moneda_salario', models.CharField(default='PEN', max_length=3, verbose_name='Moneda')),
77             ('disponibilidad', models.CharField(choices=[('inmediata', 'Inmediata'), ('2_semanas', '2 Semanas'), ('1_mes', '1 Mes'), ('negociable', 'Negociable')], default='negociable', max_length=20, verbose_name='Disponibilidad')),
78             ('portafolio_url', models.URLField(blank=True, null=True, verbose_name='URL del Portafolio')),
79             ('linkedin_url', models.URLField(blank=True, null=True, verbose_name='URL de LinkedIn')),
80             ('github_url', models.URLField(blank=True, null=True, verbose_name='URL de GitHub')),
81             ('completado', models.BooleanField(default=False, verbose_name='Perfil Completado')),
82             ('fecha_creacion', models.DateTimeField(auto_now_add=True, verbose_name='Fecha de Creación')),
83             ('fecha_actualizacion', models.DateTimeField(auto_now=True, verbose_name='Última Actualización')),
84         ],
85         options={
86             'verbose_name': 'Perfil de Postulante',
87             'verbose_name_plural': 'Perfiles de Postulantes',
88             'db_table': 'perfil_postulante',
89             'ordering': ['-fecha_actualizacion'],
90         },
91     ),
92     migrations.CreateModel(
93         name='OfertaTrabajo',
94         fields=[
95             ('id', models.BigAutoField(auto_created=True, primary_key=True, serialize=False, verbose_name='ID')),
96             ('titulo', models.CharField(max_length=200, verbose_name='Título')),
97             ('descripcion', models.TextField(verbose_name='Descripción')),
98             ('requisitos', models.TextField(blank=True, null=True, verbose_name='Requisitos')),
99             ('responsabilidades', models.TextField(blank=True, null=True, verbose_name='Responsabilidades')),
100             ('beneficios', models.TextField(blank=True, null=True, verbose_name='Beneficios')),
101             ('salario_min', models.DecimalField(blank=True, decimal_places=2, max_digits=10, null=True, verbose_name='Salario Mínimo')),
102             ('salario_max', models.DecimalField(blank=True, decimal_places=2, max_digits=10, null=True, verbose_name='Salario Máximo')),
103             ('moneda', models.CharField(default='PEN', max_length=3, verbose_name='Moneda')),
104             ('ubicacion', models.CharField(blank=True, max_length=200, null=True, verbose_name='Ubicación')),
105             ('modalidad', models.CharField(choices=[('presencial', 'Presencial'), ('remoto', 'Remoto'), ('hibrido', 'Híbrido')], max_length=20, verbose_name='Modalidad')),
106             ('tipo_contrato', models.CharField(choices=[('tiempo_completo', 'Tiempo Completo'), ('medio_tiempo', 'Medio Tiempo'), ('por_proyecto', 'Por Proyecto')],
```



```

107         ('freelance', 'Freelance'), ('practicar', 'Prcticas'), ('temporal', 'Temporal']], max_length=20, verbose_name='Tipo de Contrato')),
108         ('nivel_experiencia', models.CharField(choices=[('sin_experiencia', 'Sin Experiencia'), ('junior', 'Junior'), ('semi_senior', 'Semi Senior'), ('senior',
109         'Senior'), ('lead', 'Lead/Experto')], max_length=20, verbose_name='Nivel de Experiencia Requerido')),
110         ('vacantes_disponibles', models.IntegerField(default=1, validators=[django.core.validators.MinValueValidator(1)], verbose_name='Vacantes Disponibles')),
111         ('fecha_publicacion', models.DateTimeField(blank=True, null=True, verbose_name='Fecha de Publicacin')),
112         ('fecha_expiracion', models.DateTimeField(blank=True, null=True, verbose_name='Fecha de Expiracin')),
113         ('fecha_inicio_deseada', models.DateField(blank=True, null=True, verbose_name='Fecha de Inicio Deseada')),
114         ('estado', models.CharField(choices=[('borrador', 'Borrador'), ('pendiente_aprobacion', 'Pendiente de Aprobacin'), ('activa', 'Activa'), ('pausada', '
115         Pausada'), ('expirada', 'Expirada'), ('cerrada', 'Cerrada')], default='borrador', max_length=30, verbose_name='Estado')),
116         ('aprobada_admin', models.BooleanField(default=False, verbose_name='Aprobada por Admin')),
117         ('fecha_aprobacion', models.DateTimeField(blank=True, null=True, verbose_name='Fecha de Aprobacin')),
118         ('vistas', models.IntegerField(default=0, verbose_name='Nmero de Vistas')),
119         ('fecha_creacion', models.DateTimeField(auto_now_add=True, verbose_name='Fecha de Creacin')),
120         ('fecha_actualizacion', models.DateTimeField(auto_now=True, verbose_name='ltima Actualizacin')),
121         ('categoria', models.ForeignKey(null=True, on_delete=django.db.models.deletion.SET_NULL, related_name='ofertas', to='MyWebApps.categoria', verbose_name=
122         'Categoria')),
123         ('empresa', models.ForeignKey(on_delete=django.db.models.deletion.CASCADE, related_name='ofertas', to='MyWebApps.empresa', verbose_name='Empresa')),
124     ],
125     options={
126         'verbose_name': 'Oferta de Trabajo',
127         'verbose_name_plural': 'Ofertas de Trabajo',
128         'db_table': 'oferta_trabajo',
129         'ordering': ['-fecha_publicacion', '-fecha_creacion'],
130     },
131 ),
132 migrations.CreateModel(
133     name='Postulacion',
134     fields=[
135         ('id', models.BigAutoField(auto_created=True, primary_key=True, serialize=False, verbose_name='ID')),
136         ('fecha_postulacion', models.DateTimeField(auto_now_add=True, verbose_name='Fecha de Postulacin')),
137         ('estado', models.CharField(choices=[('pendiente', 'Pendiente'), ('en_revision', 'En Revisin'), ('preseleccionado', 'Preseleccionado'), ('entrevista', '
138         En Entrevista'), ('rechazado', 'Rechazado'), ('aceptado', 'Aceptado')], default='pendiente', max_length=20, verbose_name='Estado')),
139         ('carta_presentacion', models.TextField(blank=True, null=True, verbose_name='Carta de Presentacin')),
140         ('cv_url_postulacion', models.CharField(blank=True, max_length=500, null=True, verbose_name='URL del CV')),
141         ('fecha_cambio_estado', models.DateTimeField(blank=True, null=True, verbose_name='Fecha de Cambio de Estado')),
142         ('notas_empleador', models.TextField(blank=True, null=True, verbose_name='Notas del Empleado')),
143         ('puntuacion_match', models.IntegerField(blank=True, null=True, validators=[django.core.validators.MinValueValidator(0), django.core.validators.
144         MaxValueValidator(100)], verbose_name='Puntuacin de Match (%)')),
145         ('oferta', models.ForeignKey(on_delete=django.db.models.deletion.CASCADE, related_name='postulaciones', to='MyWebApps.ofertatrabajo', verbose_name='
146         Oferta de Trabajo')),
147         ('postulante', models.ForeignKey(on_delete=django.db.models.deletion.CASCADE, related_name='postulaciones', to='MyWebApps.perfilpostulante',
148         verbose_name='Postulante')),
149     ],
150     options={
151         'verbose_name': 'Postulacin',
152         'verbose_name_plural': 'Postulaciones',
153         'db_table': 'postulacion',
154         'ordering': ['-fecha_postulacion'],
155     },
156 ),
157 migrations.CreateModel(
158     name='Usuario',
159     fields=[
160         ('id', models.BigAutoField(auto_created=True, primary_key=True, serialize=False, verbose_name='ID')),
161         ('password', models.CharField(max_length=128, verbose_name='password')),
162         ('last_login', models.DateTimeField(blank=True, null=True, verbose_name='last login')),
163         ('is_superuser', models.BooleanField(default=False, help_text='Designates that this user has all permissions without explicitly assigning them.',
164         verbose_name='superuser status')),
165         ('email', models.EmailField(max_length=254, unique=True, verbose_name='Correo Electrónico')),
166         ('nombre', models.CharField(max_length=100, verbose_name='Nombre')),
167         ('apellido', models.CharField(blank=True, max_length=100, verbose_name='Apellido')),
168         ('telefono', models.CharField(blank=True, max_length=20, null=True, verbose_name='Teléfono')),
169         ('tipo_usuario', models.CharField(choices=[('postulante', 'Postulante'), ('empleador', 'Empleador'), ('admin', 'Administrador')], default='postulante',
170         max_length=20, verbose_name='Tipo de Usuario')),
171         ('estado', models.CharField(choices=[('activo', 'Activo'), ('inactivo', 'Inactivo'), ('suspendido', 'Suspendido')], default='activo', max_length=20,
172         verbose_name='Estado')),
173         ('email_verificado', models.BooleanField(default=False, verbose_name='Email Verificado')),
174         ('fecha_creacion', models.DateTimeField(auto_now_add=True, verbose_name='Fecha de Creacin')),
175         ('fecha_actualizacion', models.DateTimeField(auto_now=True, verbose_name='ltima Actualizacin')),
176         ('is_active', models.BooleanField(default=True)),
177         ('is_staff', models.BooleanField(default=False)),
178         ('groups', models.ManyToManyField(blank=True, help_text='The groups this user belongs to. A user will get all permissions granted to each of their
179         groups.', related_name='user_set', related_query_name='user', to='auth.group', verbose_name='groups')),
180         ('user_permissions', models.ManyToManyField(blank=True, help_text='Specific permissions for this user.', related_name='user_set', related_query_name='
181         user', to='auth.permission', verbose_name='user permissions')),
182     ],
183     options={
184         'verbose_name': 'Usuario',
185         'verbose_name_plural': 'Usuarios',
186         'db_table': 'usuario',
187         'ordering': ['-fecha_creacion'],
188     },
189 ),
190 migrations.AddField(
191     model_name='perfilpostulante',
192     name='usuario',
193     field=models.OneToOneField(on_delete=django.db.models.deletion.CASCADE, related_name='perfil_postulante', to=settings.AUTH_USER_MODEL, verbose_name='Usuario
194     '),
195 ),
196 migrations.CreateModel(
197     name='Notificacion',
198     fields=[
199         ('id', models.BigAutoField(auto_created=True, primary_key=True, serialize=False, verbose_name='ID')),
200         ('tipo', models.CharField(choices=[('postulacion', 'Nueva Postulacin'), ('estado_postulacion', 'Cambio Estado Postulacin'), ('nueva_oferta', 'Nueva
201         Oferta'), ('mensaje', 'Mensaje'), ('alerta', 'Alerta'), ('sistema', 'Sistema')], max_length=30, verbose_name='Tipo')),
202         ('titulo', models.CharField(max_length=200, verbose_name='Ttulo')),
203         ('mensaje', models.TextField(verbose_name='Mensaje')),
204         ('enlace', models.CharField(blank=True, max_length=500, null=True, verbose_name='Enlace')),
205         ('leida', models.BooleanField(default=False, verbose_name='Leda')),
206         ('fecha_creacion', models.DateTimeField(auto_now_add=True, verbose_name='Fecha de Creacin')),
207         ('fecha_leida', models.DateTimeField(blank=True, null=True, verbose_name='Fecha de Lectura')),
208         ('usuario', models.ForeignKey(on_delete=django.db.models.deletion.CASCADE, related_name='notificaciones', to=settings.AUTH_USER_MODEL, verbose_name='
209         Usuario')),

```

```
194     ],
195     options={
196         'verbose_name': 'Notificacin',
197         'verbose_name_plural': 'Notificaciones',
198         'db_table': 'notificacion',
199         'ordering': ['-fecha_creacion'],
200     },
201 ),
202 migrations.CreateModel(
203     name='Favorito',
204     fields=[
205         ('id', models.BigAutoField(auto_created=True, primary_key=True, serialize=False, verbose_name='ID')),
206         ('fecha_agregado', models.DateTimeField(auto_now_add=True, verbose_name='Fecha Agregado')),
207         ('oferta', models.ForeignKey(on_delete=django.db.models.deletion.CASCADE, related_name='favoritos', to='MyWebApps.ofertatrabajo', verbose_name='Oferta')
208             → ),
209         ('usuario', models.ForeignKey(on_delete=django.db.models.deletion.CASCADE, related_name='favoritos', to=settings.AUTH_USER_MODEL, verbose_name='Usuario'
210             → )),
211     ],
212     options={
213         'verbose_name': 'Favorito',
214         'verbose_name_plural': 'Favoritos',
215         'db_table': 'favorito',
216         'ordering': ['-fecha_agregado'],
217     },
218 ),
219 migrations.AddField(
220     model_name='empresa',
221     name='usuario',
222     field=models.OneToOneField(on_delete=django.db.models.deletion.CASCADE, related_name='empresa', to=settings.AUTH_USER_MODEL, verbose_name='Usuario'),
223 ),
224 migrations.AddIndex(
225     model_name='ofertatrabajo',
226     index=models.Index(fields=['estado', 'fecha_publicacion'], name='oferta_trab_estado_2114ed_idx'),
227 ),
228 migrations.AddIndex(
229     model_name='ofertatrabajo',
230     index=models.Index(fields=['categoria', 'estado'], name='oferta_trab_categor_b7324a_idx'),
231 ),
232 migrations.AddIndex(
233     model_name='ofertatrabajo',
234     index=models.Index(fields=['modalidad', 'estado'], name='oferta_trab_modalid_af05a7_idx'),
235 ),
236 migrations.AddIndex(
237     model_name='postulacion',
238     index=models.Index(fields=['estado', 'fecha_postulacion'], name='postulacion_estado_800a96_idx'),
239 ),
240 migrations.AddIndex(
241     model_name='postulacion',
242     index=models.Index(fields=['oferta', 'estado'], name='postulacion_oferta_6ebd41_idx'),
243 ),
244 migrations.AddIndex(
245     model_name='postulacion',
246     index=models.Index(fields=['postulante', 'estado'], name='postulacion_postula_91b3bc_idx'),
247 ),
248 migrations.AlterUniqueTogether(
249     name='postulacion',
250     unique_together=({'oferta', 'postulante'}),
251 ),
252 migrations.AddIndex(
253     model_name='notificacion',
254     index=models.Index(fields=['usuario', 'leida', '-fecha_creacion'], name='notificacio_usuario_59fc25_idx'),
255 ),
256 migrations.AlterUniqueTogether(
257     name='favorito',
258     unique_together=({'usuario', 'oferta'}),
259 )
```

Explicación de la migración

- La migración crea tablas con 'BigAutoField' como PK. Esto es coherente con aplicaciones modernas que esperan muchos registros.
- Se crean índices compuestos en 'oferta_trabajo' y 'postulacion' que aceleran consultas por 'estado' y 'fecha_publicacion' / 'f'.
- 'unique_together' evita duplicados lógicos en postulaciones y favoritos.
- Las relaciones 'OneToOneField' entre 'Usuario' y 'Empresa' / 'PerfilPostulante' se establecen mediante 'AddField' para respetar dependencias.

7. Mapeo Lógico → Físico

Entidades principales y columnas resultantes

- **Usuario** → tabla 'usuario' con columnas: id, email (unique), nombre, apellido, telefono, tipo_usuario, estado, email_verificado.
- **Empresa** → tabla 'empresa' con FK 'usuario_id'.
- **PerfilPostulante** → tabla 'perfil_postulante' con FK 'usuario_id'.
- **OfertaTrabajo** → tabla 'oferta_trabajo' con FK 'empresa_id' y FK 'categoria_id'. Índices en ('estado', 'fecha_publicacion').
- **Postulacion** → tabla 'postulacion' con FK 'oferta_id' y FK 'postulante_id'. Restricción única ('oferta', 'postulante').
- **Favorito y Notificacion** con sus relaciones y campos.

8. Consultas y ejemplos de uso (Django ORM)

Ejemplos útiles para el desarrollo y la defensa ante el docente

Listing 3: Ejemplos de uso en Django shell

```
1 # Crear un usuario (postulante)
2 from MyWebApps.models import Usuario, PerfilPostulante, Empresa, OfertaTrabajo, Postulacion
3 u = Usuario.objects.create_user(email="ana@example.com", password="pass123", nombre="Ana")
4
5 # Crear perfil de postulante
6 perfil = PerfilPostulante.objects.create(usuario=u, titulo_profesional="Ing. Sistemas", aos_experiencia=3)
7
8 # Crear empresa (usuario empleador)
9 e_user = Usuario.objects.create_user(email="rrhh@empresa.com", password="pass123", nombre="RRHH", tipo_usuario="empleador")
10 empresa = Empresa.objects.create(usuario=e_user, nombre_empresa="ACME S.A.", ruc="12345678901")
11
12 # Publicar oferta
13 of = OfertaTrabajo.objects.create(empresa=empresa, titulo="Desarrollador Backend", descripcion="Python/Django", modalidad="remoto", tipo_contrato="tiempo_completo")
14
15 # Postular a oferta
16 post = Postulacion.objects.create(oferta=of, postulante=perfil, carta_presentacion="Interesado en el puesto")
17
18 # Consultas frecuentes
19 OfertaTrabajo.objects.filter(estado="activa").order_by("-fecha_publicacion")[:10] # listado principales
20 Postulacion.objects.filter(postulante=perfil).select_related("oferta") # postulaciones de un postulante
```

9. Decisiones de diseño y justificación técnica

- **Email como USERNAME_FIELD**: mejora UX y unicidad de identidad. UsuarioManager implementado para crear superusuarios.
- **Campos URL para CV/Logo**: evita almacenamiento de binarios en BD. Permite integrarse con S3 u otros storage.
- **Índices compuestos**: elegidos según filtros esperados (estado + fecha). Reduce escaneo de tabla en listados públicos.
- **On delete cascade**: se elimina información dependiente al borrar empresa o usuario. Considerar soft-delete si se requiere recuperación.

- **Validadores en años_experiencia y puntuacion_match:** mantienen integridad semántica a nivel aplicación.
- **Campos fechas auto_add/autoupdate:** útil para auditoría básica. Para auditoría avanzada usar modelo audit/log separado.

10. Riesgos, mejoras y recomendaciones

Riesgos detectados

- Eliminación física (cascade) puede borrar datos valiosos. Evaluar soft delete.
- Índices aumentan el costo de escritura. Balancear si la carga de inserciones es alta.
- Campos de texto grandes para CV/experiencia reducen capacidad de búsqueda. Considerar usar tablas normalizadas o motores de búsqueda (ElasticSearch) para búsquedas por habilidades.

Mejoras recomendadas

- Implementar soft delete (campo 'deleted_at') en tablas críticas.
- Añadir métricas y telemetría para consultas frecuentes (vistas, postulaciones).
- Implementar índices parciales si la base de datos lo admite (p.ej. índices solo para estado='activa').
- Añadir triggers o jobs para limpieza/archivado de ofertas expiradas.

11. Ejecución y pruebas básicas

Comandos para levantar el proyecto (entorno local)

Listing 4: Comandos básicos Django

```
1 # crear entorno virtual (ejemplo)
2 python -m venv .venv
3 source .venv/bin/activate
4
5 # instalar dependencias (asumiendo requirements.txt)
6 pip install -r requirements.txt
7
8 # aplicar migraciones
9 python manage.py migrate
10
11 # crear superusuario
12 python manage.py createsuperuser --email admin@example.com
13
14 # levantar servidor de desarrollo
15 python manage.py runserver
```

Pruebas rápidas (curl / httpie)

Listing 5: Ejemplos de prueba (API REST si existen vistas)

```
1 # ejemplo: listar ofertas (si existe endpoint /api/ofertas/)
2 curl -sS http://127.0.0.1:8000/api/ofertas/ | jq
3
4 # ejemplo: autenticación y creación (si API token/session)
5 # (estos endpoints dependen de la implementación de views)
```

12. Evidencias y trazabilidad

El repositorio que contiene `models.py` y `migrations/0001_initial.py` es la evidencia de la implementación. Para la defensa se recomienda:

- Mostrar ‘python manage.py showmigrations’ y ‘python manage.py sqlmigrate app 0001’ para demostrar DDL generado.
- Ejecutar ejemplos de ORM en ‘python manage.py shell’.
- Presentar las tablas resultantes en la BD (pgAdmin / psql) y explicar índices.

13. Conclusiones

- El modelo implementado cubre el dominio principal (usuarios, empresas, ofertas, postulaciones).
- Las decisiones de diseño están orientadas a consultas típicas de buscadores de empleo (filtros por estado, categoría, modalidad).
- La migración generada por Django deja la base lista para utilizar en producción con ajustes menores (backup, políticas de retención).

Ejemplo (Fetch):

```
1 fetch("/api/ofertas/")
2 .then(r => r.json())
3 .then(data => console.log(data))
4 .catch(err => console.error(err));
```

14. Calificación

Cuadro 1: Rúbrica para tipo de Informe

Informe	Descripción	Cumple	No cumple
L ^A T _E X	El informe está en formato PDF desde L ^A T _E X, con un formato limpio (buena presentación) y fácil de leer.	20	0
MarkDown	El informe está en formato PDF desde MarkDown README.md, con un formato limpio (buena presentación) y fácil de leer.	17	0
MS Word	El informe está en formato PDF desde plantilla MS Word, con un formato limpio (buena presentación) y fácil de leer.	15	0
Observaciones	Por cada observación se le descontará puntos.	-	-

Referencias

- Documentación oficial Django: <https://docs.djangoproject.com/>